



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|-------------------|-------------|----------------------|---------------------|------------------|
| 10/726,857 | 12/02/2003 | Paul Thomas Knowles | 1801270.00132US1 | 5597 |
| 23483 | 7590 | 03/24/2009 | | |
| WILMERHALE/BOSTON | | | | |
| 60 STATE STREET | | | | |
| BOSTON, MA 02109 | | | | |
| EXAMINER | | | | |
| WU, JUNCHUN | | | | |
| ART UNIT | | PAPER NUMBER | | |
| 2191 | | | | |
| NOTIFICATION DATE | | DELIVERY MODE | | |
| 03/24/2009 | | ELECTRONIC | | |

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

michael.mathewson@wilmerhale.com

teresa.carvalho@wilmerhale.com

sharon.mathews@wilmerhale.com

Office Action Summary

Application No.

10/726,857

Applicant(s)

KNOWLES, PAUL THOMAS

Examiner

JUNCHUN WU

Art Unit

2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 30 December 2008.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-8, 11, 12, 14 and 19-30 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-7, 12, 14 and 19-30 is/are rejected.
- 7) ☒ Claim(s) 8 and 11 is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date 12/30/2008.
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____.
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. This office action is in response to RCE filed on Dec. 30, 2008.
2. Claim 1 has been amended.
3. Claims 1-8, 11, 12, 14 and 19-30 are pending in this application.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 1-5 and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over by Babaian et al. (US Pub. No. 20020092002 A1 hereinafter "Babaian"), in view of Buzbee (US Patent. No. 6,219,832 B1 hereinafter "Buzbee")

Per claim 1 (Currently Amended)

Babaian discloses

A method of handling exceptions encountered during the translation of subject program code into target code, comprising:

- detecting the occurrence of an exception ([0007] "*the exception handler initiates a process designed to provide a solution to the problem or to the change in conditions whenever encountered during the execution of the program. Exceptions are synchronous events that are generated in response to certain conditions detected during the execution of an instruction.*").

- selecting a level of subject context precision required for the detected exception from a plurality of possible levels of precision ([0084]).

Further, Babaian discloses handling precise exceptions in a computer system that executes a plurality of operations in parallel (e.g. see [Abstract]), but does not explicitly disclose

- invoking a signal handler to handle the detected exception using the selected level of precision,
- the selected level of subject context precision comprises a level in which less than an entire subject processor state is passed to the signal handler,
- and the selected level of subject context precision that is passed to the signal handler comprises one or more of (i) a last known stack frame, (ii) no subject processor state and (iii) a precise program counter value.

However, Buzbee discloses

- invoking a signal handler to handle the detected exception using the selected level of precision (col.2 lines 55-60 *"The DTS intercepts all system calls made by the application. If a signal call does not arm an exception handler, then the DTS continues to execute the application in aggressive mode. If, however, the signal call arms an exception handler, then the DTS looks to the annotations to determine the number of formal parameters expected by the handler. "*)
- the selected level of subject context precision comprises a level in which less than an entire subject processor state is passed to the signal handler (col.2 lines 61-67 *"The number of formal parameters is relevant because, when an exception occurs, the machine state is the third formal parameter passed with the exception signal. Therefore, if an*

exception handler expects two or fewer parameters, then that handler does not use the machine state. Conversely, if a handler expects three or more parameters, then that handler may use the machine state.”)

- and the selected level of subject context precision that is passed to the signal handler comprises one or more of (i) a last known stack frame, (ii) no subject processor state and (iii) a precise program counter value (col.1 lines 34-43 “*When executing the application, the dynamic translator must accurately simulate the entire state and actions of the emulated computer system. Signal handling is one of the most difficult aspects of the computer system to simulate. Normally, if an application were executing and a signal arrived, then the operating system would create a snapshot of the machine state of the application as it existed when the signal arrived. This snapshot contains the values of all registers, the program counter, and the status bits. In short, the snapshot contains all visible CPU state information.*”)
- Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Babaian’s teachings by adding invoking a signal handler to handle the detected exception using the selected level of precision, the selected level of subject context precision comprises a level in which less than an entire subject processor state is passed to the signal handler, and the selected level of subject context precision that is passed to the signal handler comprises one or more of (i) a last known stack frame, (ii) no subject processor state and (iii) a precise program counter value by Buzbee in order to improved handling of exceptions in the dynamic translator properly handles

Art Unit: 2191

signals yet does not introduce unnecessary instructions for materializing the machine state (Buzbee col.3 lines 10-14).

Per claim 2 (Original)

the rejection of claim 1 is incorporated and Babaian further discloses

- the exception occurrence detecting step detects the occurrence of an exception signal during translation of the subject program code ([0012] *“Binary translation means that a foreign code is processed by host software to produce new host code corresponding to the foreign code. The new host code is called binary translated code. To preserve foreign code behavior in binary translated code on the host computer system, the system must maintain precise exceptions.”*).

Per claim 3 (Original)

the rejection of claim 2 is incorporated and Babaian further discloses

- the target code generated by the translation invokes a proxy signal handler to handle the detected exception ([0009]).

Per claim 4 (Original)

the rejection of claim 1 is incorporated and Babaian further discloses

- the exception occurrence detecting step detects the occurrence of an exception signal during execution of the target code ([0012] *“the present invention also provides a*

technique for maintaining precise exceptions in binary translated code executing on a host platform having explicit parallelism architecture”).

Per claim 5 (Original)

the rejection of claim 4 is incorporated and Babaian further discloses

- target operating system invokes a proxy signal handler to handle the detected exception.
([0008] *“As the exception occurs, the exception handler in the operating system will load the needed page from disk into the memory and then re-execute faulted user's instruction.”*).

6. Claims 6 and 7 are rejected under 35 U.S.C. 103(a) as being unpatentable over by Babaian, in view of Buzbee and further view of Gokingco et al. (“Porting software from an MC68040 to an MC68060”, April 6, 1995, hereinafter “Gokingco”).

Per claim 6 (Original)

the rejection of claim 1 is incorporated

Babaian, and Buzbee does not explicitly discloses

- the default level of subject context precision is a last known stack frame

But Gokingco discloses

- the default level of subject context precision is a last known stack frame (Sec. 3.0, 3rd paragraph; when an exception occurs, the stack frame was created by system as default level).

Art Unit: 2191

- Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine teachings of Babaian and Buzbee by adding the default level of subject context precision is a last known stack frame by Gokingco in order to provide information to the exception and to provide a way to return to normal processing (Gokingco, Sec. 3.0, 4th paragraph lines 1-2).

Per claim 7 (Original)

the rejection of claim 6 is incorporated

Babaian and Buzbee does not explicitly disclose

- the last known stack frame includes a last known stack pointer value, a base pointer value, and a program counter register value,

But Gokingco discloses

- the last known stack frame includes a last known stack pointer value, a base pointer value (Sec. 3.0, 5th paragraph lines 1-5).
- Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine teachings of Babaian and Buzbee by adding the last known stack frame includes a last known stack pointer value, a base pointer value by Gokingco in order to distinguish among different stack frame types and values (Gokingco, Sec. 3.0, 5th paragraph lines 6-7).

7. Claims 12 and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over by Krishnaswamy (US Pub. No. 20010013118 A1 hereinafter “Krishnaswamy”), and in view of Buzbee.

Per claims 12 and 30

Krishnaswamy discloses

In a method of handling subject code exceptions in a translation system employing a translator to translate subject code to target code, the steps comprising:

- generating a target context ([0025]).
- reconstructing a subject context using said target context, thereby generating a reconstructed subject context ([0031]).
- executing a translated version of a subject signal handler associated with a particular said exception using the reconstructed subject context, wherein the step of reconstructing a subject context comprises reconstructing less than an entire subject processor state ([0035] “*Once the interpreter reaches a GAR point, all of the asynchronous exceptions will be handled, i.e., the application context will be reconstructed and control will be transferred to the respective asynchronous exception handlers.*”)
- and the reconstructed subject context includes one or more of (i) a last known stack frame, (ii) no subject processor state and (iii) a precise program counter value ([0008] “*Various approaches have been used in conjunction with dynamic translation execution to ensure that the reconstructed application context (i.e., the hardware register state and the instruction pointer) is correct when control is transferred to an asynchronous exception handler.*”)

8. Claims 14, 19-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over by Krishnaswamy, in view of Buzbee, and further view of Gokingco.

Per claim 14 (Original)

the rejection of claim 12 is incorporated

Both Krishnaswamy and Buzbee do not teach

- step of reconstructing a subject context comprises selecting one of a plurality of subject context precision levels for processing said exception,

But Gokingco teaches

- step of reconstructing a subject context comprises selecting one of a plurality of subject context precision levels for processing said exception (Sec. 3.0, 4th paragraph lines 2-6).
- Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Krishnaswamy and Buzbee teachings by adding step of reconstructing a subject context comprises selecting one of a plurality of subject context precision levels for processing said exception by Gokingco in order to using v-table to dispatch each exception types to actual exception handler (Gokingco, Sec. 3.0, 4th paragraph).

Per claim 19 (Original)

the rejection of claim 14 is incorporated and Krishnaswamy further discloses

- step of reconstructing a subject context is performed by proxy signal handler code ([0034] “*When the interpreter 27 determines that a GAR point has been reached in the*

Art Unit: 2191

execution stream, the interpreter 27 transfers control to the dynamic translation system controller 28 which reconstructs the application context and transfers control to the asynchronous exception handler”).

Per claim 20 (Original)

the rejection of claim 19 is incorporated and Krishnaswamy further discloses

- proxy signal handler code is registered in the target code by said translator and wherein said translator further raises a flag to said proxy signal handler indicating which of said plurality of subject context precision levels is to be used in response to said particular exception ([0009]).

Per claim 21 (Original)

the rejection of claim 19 is incorporated and Krishnaswamy further discloses

- particular exception is detected during decoding of the subject code by said translator ([0003] “*When an asynchronous exception occurs, the dynamic translation system must simulate, during the execution of the translated application code, the handling of the asynchronous exception by the input application code.*”).

Per claim 22 (Original)

the rejection of claim 21 is incorporated and Krishnaswamy further discloses

- translator responds to detection of said particular exception during decoding to plant target code which generates said target context and invokes operation of the proxy signal

handler code ([0010]) “*All of the prior approaches for handling asynchronous exceptions, including the approach disclosed in the Buzbee patent, detect when an asynchronous exception has occurred, record the asynchronous exception, continue execution until a GAR point is reached, and transfer control to the asynchronous exception handler for servicing the asynchronous event when the GAR point is reached.*”).

Per claim 23 (Original)

the rejection of claim 19 is incorporated and Krishnaswamy further discloses

- particular exception arises during execution of said target code ([0029]).

Per claim 24 (Original)

the rejection of claim 23 is incorporated and Krishnaswamy further discloses

- a target operating system responds to occurrence of said particular exception during execution of said target code to pass target context to said proxy signal handler code ([0010]).

Per claim 25 (Original)

the rejection of claim 24 is incorporated and Krishnaswamy further discloses

- after receiving said target context, said proxy signal handler code calls the translator, which then invokes a selected translated subject signal handler ([0034] 4-8).

Per claim 26 (Original)

the rejection of claim 14 is incorporated

Both Krishnaswamy and Luch do not disclose

- exception is caused by one of a plurality of asynchronous external events and wherein said exception is handled using a selected default level of precision assigned to all asynchronous events.

But Gokingo discloses

- exception is caused by one of a plurality of asynchronous external events (Sec. 3.0, 2nd paragraph lines 1-3) and wherein said exception is handled using a selected default level of precision assigned to all asynchronous events (Sec. 3.0, 6th paragraph lines 1-3, using stack frame as a default level to determine the cause of the exception and handle the exception).
- Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Krishnaswamy and Buzbee teachings by adding exception is caused by one of a plurality of asynchronous external events and wherein said exception is handled using a selected default level of precision assigned to all asynchronous events by Gokingo in order to provide information to the exception and to provide a way to return to normal processing (Gokingo, Sec. 3.0, 4th paragraph lines 1-2).

Per claim 27 (Original)

the rejection of claim 26 is incorporated

Krishnaswamy and Buzbee do not disclose

- selected default level is a last known stack frame

but Gokingco discloses

- selected default level is a last known stack frame (Sec. 3.0, 3rd paragraph; when an exception occurs, the stack frame was created by system as default level).
- Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify teachings of Krishnaswamy and Buzbee by adding selected default level is a last known stack frame by Gokingco in order to provide information to the exception and to provide a way to return to normal processing (Gokingco, Sec. 3.0, 4th paragraph lines 1-2).

Per claim 28 (Original)

the rejection of claim 19 is incorporated and Krishnaswamy further discloses

- proxy signal handler code is arranged to interact with a subject register bank ([0009])

Allowable Subject Matter

9. Claims 8 and 11 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Response to Arguments

Applicant's arguments filed on Dec. 30, 2008 have been fully considered but they are not persuasive.

In the remarks,

(a) Regarding to independent claim 1, the cited art of Babaian and Leuh do not disclose “the selected level of subject context precision comprises a level in which less than an entire subject processor state is passed to the signal handler, and the selected level of subject context precision that is passed to the signal handler comprises one or more of (i) a last known stack frame, (ii) no subject processor state and (iii) a precise program counter value.”

(b) Regarding to independent claims 11 and 30, the cited art of Krishnaswamy do not teach or suggest “wherein the step of reconstructing a subject context comprises reconstructing less than an entire subject processor state and the reconstructed subject context includes one or more of (i) a last known stack frame, (ii) no subject processor state and (iii) a precise program counter value”

Examiner's response:

Examiner disagrees.

(a) Applicant's arguments with respect to claim 1 has been considered but are moot in view of the new ground(s) of rejection - see Buzbee, arts made of record, as applied hereto.

(b) Krishnaswamy discloses wherein the step of reconstructing a subject context comprises reconstructing less than an entire subject processor state ([0035] “*Once the interpreter reaches a GAR point, all of the asynchronous exceptions will be handled, i.e., the application context will be reconstructed and control will be transferred to the respective asynchronous exception handlers.*” & [0037] “*the term "GAR point" is meant to denote a location in the translated code*

where state can be recovered, i.e., where the application context can be correctly reconstructed.”

Also refer to Fig.4) and the reconstructed subject context includes one or more of (i) a last known stack frame, (ii) no subject processor state and (iii) a precise program counter value ([0008] “*Various approaches have been used in conjunction with dynamic translation execution to ensure that the reconstructed application context (i.e., the hardware register state and the instruction pointer) is correct when control is transferred to an asynchronous exception handler.*”)

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JUNCHUN WU whose telephone number is (571)270-1250. The examiner can normally be reached on 8:00-17:00 M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JW
/Wei Y Zhen/

Supervisory Patent Examiner, Art Unit 2191